Chapter 24
Configure Multilink and Link Services Interfaces

The Multilink Protocol (MP) enables you to split, recombine, and sequence datagrams across multiple logical data links. The goal of multilink operation is to coordinate multiple independent links between a fixed pair of systems, providing a virtual link with greater bandwidth than any of the members.

The JUNOS software supports two MP-based services PICs: the Multilink Services PIC and the Link Services PIC. The Multilink Services PIC and the Links Services PIC support the following MP encapsulation types at the logical unit level:

- Multilink Point-to-Point Protocol (MLPPP)
- Multilink Frame Relay (MLFR FRF.15)

The Link Services PIC also supports the Multilink Frame Relay User-to-Network Interface (UNI) and the Network-to-Network Interface (NNI) (MLFR FRF.16) encapsulation type at the physical interface level.

MLPPP and MLFR (FRF.15) are supported on interface types ml-*fpc*/*pic*/*port* and ls-*fpc*/*pic*/*port*. For MLFR (FRF.15), multiple permanent virtual circuits (PVCs) are combined into one aggregated virtual circuit (AVC). This provides fragmentation over multiple PVCs on one end and reassembly of the AVC on the other end.

MLFR (FRF.16) is supported on a channelized interface, ls-*fpc*/*pic*/*port*:*channel*, which denotes a single MLFR (FRF.16) bundle. For MLFR (FRF.16), multiple links are combined to form one logical link. Packet fragmentation and reassembly occur on a per-VC basis. Each bundle can support multiple VCs. The physical connections must be E1, T1, Channelized DS-3 to DS-1, Channelized DS-3 to DS-0, Channelized E1, Channelized STM-1, or Channelized QPP interfaces.

The standards for MLPPP, MLFR FRF.15, and MLFR FRF.16 are defined in the following specifications:

- RFC 1990, *The PPP Multilink Protocol (MP)*
- FRF.15, *End-to-End Multilink Frame Relay Implementation Agreement*
- FRF.16, *Multilink Frame Relay UNI/NNI Implementation Agreement*

To configure multilink and link services interface properties, include the ml-*fpc/pic/port* or ls-*fpc/pic/port* statement at the [edit interfaces] hierarchy level:

```
[edit interfaces]
(ml-fpc/pic/port | ls-fpc/pic/port) {
    unit logical-unit-number {
        dlci dlci-identifier;
        drop-timeout milliseconds;
        encapsulation type;
        fragment-threshold bytes;
        interleave-fragments;
        minimum-links number;
        mrru bytes;
        multicast-dlci dlci-identifier;
        short-sequence;
        family family {
            address address {
                destination address;
            }
            bundle (ml-fpc/pic/port | ls-fpc/pic/port);
            }
        }
    }
}
```

> **Note**
>
> The Multilink Services PIC is not compatible with the M40e, M160, or T-series platforms.
>
> The Link Services PIC is not compatible with the M160 or T-series platforms.

To configure link services physical interface properties, include the mlfr-uni-nni-bundle-options statement at the [edit interfaces ls-*fpc/pic/port*] hierarchy level:

```
[edit interfaces ls-fpc/pic/port]
encapsulation type;
mlfr-uni-nni-bundle-options {
    acknowledge-retries number;
    acknowledge-timer milliseconds;
    action-red-differential-delay (disable-tx | remove-link);
    drop-timeout milliseconds;
    feac-loop-respond bytes;
    hello-timer milliseconds;
    lmi-type (ansi | itu);
    minimum-links number;
    mrru bytes;
    n391 number;
    n392 number;
    n393 number;
    red-differential-delay milliseconds;
    t391 number;
    t392 number;
    yellow-differential-delay milliseconds;
}
```

This chapter is organized as follows:

For examples of multilink and link services interface configuration, see the following sections:

Configure Multilink and Link Services Logical Interface Properties

You configure multilink and link services interface properties at the logical unit level. Default settings for multilink and link services logical interface properties are described in the following section:

You can configure the following multilink and link services logical interface properties:

For general information about logical unit properties, see “Configure Logical Interface Properties” on page 63. For general information about family inet properties, see “Configure Protocol Family and Address Interface Properties” on page 73. For information about multilink and link services properties you configure at the family inet hierarchy level, see “Configure Multilink and Link Services Bundles” on page 284.

Default Settings for Multilink and Link Services Logical Interfaces

Table 22 lists the default settings for multilink and link services statements, together with the other permitted values or value ranges.

Table 22: Multilink and Link Services Logical Interface Statements

Option	Default Value	Possible Values
DLCI (data-link connection identifier)	None	16 through 1022
Drop timeout period	0 ms	0 through 2000 ms
Encapsulation	For multilink interfaces, multilink-ppp. For link services interfaces, multilink-frame-relay-end-to-end.	multilink-frame-relay-end-to-end, multilink-ppp
Fragmentation threshold	0 bytes	128 through 16,320 bytes (*N*x64)
Interleave fragments	disabled	enabled, disabled
Minimum links	1 link	1 through 8 links
MRRU (maximum received reconstructed unit)	1524 bytes	1500 through 4500 bytes
Sequence ID format for ML-PPP	24 bits	12 or 24 bits
Sequence ID format for MLFR (FRF.15 and FRF.16)	12 bits	12 bits

See Table 23 on page 280 for statements that apply to link services physical interfaces only.

Configure a Link Services Point-to-Point DLCI

For link services interfaces only, you can configure multiple DLCIs for each MLFR (FRF.16) or MLPPP bundle. A channelized interface, such as ls-1/1/1:0, denotes a single MLFR (FRF.16) bundle. To configure a DLCI, include the dlci statement at the [edit interfaces *interface-name* unit *logical-unit-number*] hierarchy level:

```
[edit interfaces interface-name unit logical-unit-number]
dlci dlci-identifier;
```

The DLCI identifier is a value from 16 through 1022. Numbers 1 through 15 are reserved for future use. A point-to-point interface can have one DLCI.

When you configure point-to-point connections, the MTU sizes on both sides of the connection must be the same.

DLCIs are not supported on multilink interfaces.

Configure a Link Services Multicast-Capable DLCI

For link services interfaces only, you can configure multiple multicast-capable DLCIs for each MLFR (FRF.16) bundle. A channelized interface, such as ls-1/1/1:0, denotes a single MLFR (FRF.16) bundle. By default, Frame Relay connections assume unicast traffic. If your Frame Relay switch performs multicast replication, you can configure the link services connection to support multicast traffic by including the multicast-dlci statement at the [edit interfaces *interface-name* unit *logical-unit-number*] hierarchy level:

```
[edit interfaces interface-name unit logical-unit-number]
multicast-dlci dlci-identifier;
```

The DLCI identifier is a value from 16 through 1022 that defines the Frame Relay DLCI over which the switch expects to receive multicast packets for replication.

You can configure multicast support only on point-to-multipoint link services connections. Multicast-capable DLCIs are not supported on multilink interfaces.

If keepalives are enabled, causing the interface to send Local Management Interface (LMI) messages during idle times, the number of possible DLCI configurations is limited by the maximum transmission unit (MTU) selected for the interface. For more information, see “Configure Link Services Keepalive Settings on Frame Relay LMI” on page 282.

Configure a Multilink and Link Services Drop Timeout Period

By default, the drop timeout parameter is disabled. You can configure a drop timeout value to provide a recovery mechanism if individual links in the multilink or link services bundle drop one or more packets. Make sure the value you set is larger than the expected differential delay across the links, although drop timeout is not a differential delay tolerance setting, and does not limit the overall latency. You can configure differential delay tolerance for link services interfaces only. For more information, see “Configure Link Services Differential Delay” on page 281.

To configure the drop timeout value, include the drop-timeout statement at the [edit interfaces ml-*fpc/pic/port* unit *logical-unit-number*] or [edit interfaces ls-*fpc/pic/port* unit *logical-unit-number*] hierarchy level:

```
drop-timeout milliseconds;
```

For link services interfaces, you also can configure the drop timeout value at the physical interface level also by including the drop-timeout statement at the [edit interfaces ls-*fpc/pic/port* mlfr-uni-nni-bundle-options] hierarchy level:

```
[edit interfaces ls-fpc/pic/port mlfr-uni-nni-bundle-options]
drop-timeout milliseconds;
```

The drop timer has the duration of 0 through 2000 ms. Values less than 5 ms are not recommended; a value of 0 disables the timer.

Note

For multilink or link services interfaces, if a packet or fragment encounters an error condition and is destined for a disabled bundle or link, it does not contribute to the dropped packet and frame counts in the per-bundle statistics. The packet is counted under the global error statistics and is not included in the global output bytes and output packet counts. This unusual accounting happens only if the error conditions are generated inside the multilink interface, not if the packet encounters errors on the wire or elsewhere in the network.

Configure Multilink and Link Services Logical Interface Encapsulation

Multilink and Link Services interfaces support the following logical interface encapsulation types:

MLPPP

MLFR End-to-End

By default, the logical interface encapsulation type on multilink interfaces is MLPPP. The default logical interface encapsulation type on link services interfaces is MLFR End-to-End. For more information, see “Configure the Encapsulation on a Logical Interface” on page 70.

You can configure physical interface encapsulation on link services interfaces. For more information, see “Configure Link Services Physical Interface Encapsulation” on page 280.

To configure multilink or link services encapsulation, include the encapsulation statement at the [edit interfaces ml-*fpc/pic/port* unit *logical-unit-number*] or [edit interfaces ls-*fpc/pic/port* unit *logical-unit-number*] hierarchy level:

```
encapsulation type;
```

You must also configure the T1 or E1 physical interface with the same encapsulation type.

Configure a Multilink and Link Services Fragmentation Threshold

By default, the fragmentation threshold parameter is disabled. For interfaces with MLPPP encapsulation only, you can configure a fragmentation threshold to set a maximum size for packet payloads transmitted across the individual links within the multilink circuit. The software splits any incoming packet that exceeds the fragmentation threshold into smaller units suitable for the circuit size; it reassembles the fragments at the other end, but does not affect the output traffic stream. The threshold value affects the payload only; it does not affect the MLPPP header.

Note

To ensure proper load-balancing:

For MLFR (FRF.15 and FRF.16) interfaces, do not include the fragmentation-threshold statement in the configuration.

For MLPPP interfaces, do not include both the fragmentation-threshold statement and the short-sequence statement in the configuration.

For MLFR (FRF.15 and FRF.16) and MLPPP interfaces, if the MTU of links in a bundle is less than the bundle MTU plus encapsulation overhead, then fragmentation is automatically enabled. You should avoid this situation for MLFR (FRF.15 and FRF.16) interfaces and for MLPPP interfaces on which short-sequencing is enabled.

To configure a fragmentation threshold value, include the fragment-threshold statement at the [edit interfaces ml-*fpc/pic/port* unit *logical-unit-number*] or [edit interfaces ls-*fpc/pic/port* unit *logical-unit-number*] hierarchy level:

```
fragment-threshold bytes;
```

For link services interfaces, you also can configure a fragmentation threshold value at the physical interface level by including the fragment-threshold statement at the [edit interfaces ls-*fpc/pic/port* mlfr-uni-nni-bundle-options] hierarchy level:

```
[edit interfaces ls-fpc/pic/port mlfr-uni-nni-bundle-options]
fragment-threshold bytes;
```

The maximum fragment size can be from 128 through 16,320 bytes. The JUNOS software automatically subdivides packet payloads that exceed this value. Any value you set must be a multiple of 64 bytes (*N*x64). The default value, 0, results in no fragmentation.

Configure Link Services Delay-Sensitive Packet Interleaving

For link services interfaces only, you can interleave long packets with high-priority packets. This allows small delay-sensitive packets, such as Voice over IP (VoIP) packets, to interleave with long fragmented packets. This minimizes the latency of delay-sensitive packets. To interleave packets, include the interleave-fragments statement at the [edit interfaces ls-*fpc/pic/port* unit *logical-unit-number*] hierarchy level:

```
interleave-fragments;
```

By default, packet interleaving is disabled.

Configure Multilink and Link Services Minimum Links

You can set the minimum number of links that must be up for the multilink bundle as a whole to be labeled up. To set the minimum number, include the minimum-links statement at the [edit interfaces ml-*fpc/pic/port* unit *logical-unit-number*] or [edit interfaces ls-*fpc/pic/port* unit *logical-unit-number*] hierarchy level:

```
minimum-links number;
```

For link services interfaces, you also can configure the minimum number of links at the physical interface level by including the minimum-links statement at the [edit interfaces ls-*fpc/pic/port* mlfr-uni-nni-bundle-options] hierarchy level:

```
[edit interfaces ls-fpc/pic/port mlfr-uni-nni-bundle-options]
minimum-links number;
```

You can set from 1 through 8 minimum links. By default, the links minimum is 1.

Configure Multilink and Link Services MRRU

The maximum received reconstructed unit (MRRU) is similar to a maximum transmission unit (MTU), but applies only to multilink bundles; it is the maximum packet size that the multilink interface can process. By default, the MRRU is set to 1500 bytes; you can configure a different MRRU value if the peer equipment allows. The MRRU includes the original payload plus the 2-byte PPP header, but not the additional MLPPP or MLFR header applied while the individual multilink packets are traversing separate links in the bundle.

To configure a different MRRU value, include the mrru statement at the [edit interfaces ml-*fpc/pic/port* unit *logical-unit-number*] or [edit interfaces ls-*fpc/pic/port* unit *logical-unit-number*] hierarchy level:

```
mrru bytes;
```

For link services interfaces, you also can configure a different MRRU at the physical interface level by including the mrru statement at the [edit interfaces ls-*fpc/pic/port* mlfr-uni-nni-bundle-options] hierarchy level:

```
[edit interfaces ls-fpc/pic/port mlfr-uni-nni-bundle-options]
mrru bytes;
```

The MRRU size can be from 1500 through 4500 bytes.

> Note
>
> If you set the MRRU on a bundle to a value larger than the MTU of the individual links within it, you must enable a fragmentation threshold for that bundle. Set the threshold to a value no larger than the smallest MTU of any link included in the bundle.
>
> Determine the appropriate MTU size for the bundle by ensuring the MTU size does not exceed the sum of the encapsulation overhead and the MTU sizes for the links in the bundle.

Configure Multilink and Link Services Sequence Format

For MLPPP, the sequence header format is set to 24 bits by default. You can configure an alternative value of 12 bits, but 24 bits is considered the more robust value for most networks.

To configure a different sequence header value, include the short-sequence statement at the [edit interfaces ml-*fpc/pic/port* unit *logical-unit-number*] or [edit interfaces ls-*fpc/pic/port* unit *logical-unit-number*] hierarchy level:

```
short-sequence;
```

For MLFR, the sequence header format is set to 24 bits by default. This is the only valid option.

Configure Link Services Physical Interface Properties

You configure link services interface properties at the logical unit and physical interface level.

Default settings for link services physical interface properties are described in the following section:

Default Settings for Link Services Interfaces on page 280

You can configure the following link services physical interface properties:

Configure Link Services Physical Interface Encapsulation on page 280

Configure Link Services Acknowledgment Timers on page 281

Configure Link Services Differential Delay on page 281

Configure Link Services Keepalive Settings on Frame Relay LMI on page 282

For descriptions of link services physical interface properties that also can be configured at the logical unit level, see “Configure Multilink and Link Services Logical Interface Properties” on page 273.

Default Settings for Link Services Interfaces

Table 23 lists the default settings for multilink and link services statements, together with the other permitted values or value ranges.

Table 23: Link Services Physical Interface Statements

Option	Default Value	Possible Values
Action red differential delay	disable-tx	disable-tx, remove-link
Red differential delay	10	1 through 2000
Yellow differential delay	6	1 through 2000
Drop timeout period	0 ms	0 through 2000 ms
Encapsulation	multilink-frame-relay-uni-nni	multilink-frame-relay-uni-nni
Fragmentation threshold	0 bytes	128 through 16,320 bytes (*N*x64)
LMI type	itu	ansi, itu
Minimum links	1 link	1 through 8 links
MRRU (maximum received reconstructed unit)	1524 bytes	1500 through 4500 bytes
n391 (full status polling counter)	6	1 through 255
n392 (LMI error threshold)	3	1 through 10
n393 (LMI monitored event count)	4	1 through 10
t391 (link integrity verify polling timer)	10	5 through 30
t392 (polling verification timer)	15	5 through 30
Sequence ID format for MLFR	12 bits	12 bits

Configure Link Services Physical Interface Encapsulation

Link services interfaces support the physical interface encapsulation MLFR UNI NNI. By default, the physical interface encapsulation on link services interfaces is MLFR UNI NNI. Multilink interfaces do not support physical interface encapsulation.

For more information, see “Configure the Encapsulation on a Physical Interface” on page 48.

You can configure logical interface encapsulation on multilink and link services interfaces. For more information, see “Configure Multilink and Link Services Logical Interface Encapsulation” on page 276.

To explicitly configure link services physical interface encapsulation, include the encapsulation statement at the [edit interfaces ls-*fpc/pic/port*] hierarchy level:

```
[edit interfaces ls-fpc/pic/port]
encapsulation type;
```

You must also configure the T1 or E1 physical and physical interface with the same encapsulation type.

Configure Link Services Acknowledgment Timers

For link services interfaces, each link endpoint in a bundle initiates a request for bundle operation with its peer by transmitting an add link message. A hello message notifies the peer endpoint that the local endpoint is up. Both ends of a link generate a hello message periodically, or as configured with the hello timer. A remove link message notifies the peer that the local end management is removing the link from bundle operation. Endpoints respond to add link, remove link, and hello messages by sending acknowledgement messages.

You can configure the maximum period to wait for an add link acknowledgement, hello acknowledgement, or remove link acknowledgement by including the acknowledge-timer statement at the [edit interfaces ls-*fpc*/*pic*/*port* mlfr-uni-nni-bundle-options] hierarchy level:

```
[edit interfaces ls-fpc/pic/port mlfr-uni-nni-bundle-options]
acknowledge-timer milliseconds;
```

The acknowledgement timer can be from 1 through 10 milliseconds. The default is 4 milliseconds.

For link services interfaces, you can configure the number of retransmission attempts to be made for consecutive hello or remove link messages after the expiration of the acknowledgement timer by including the acknowledge-retries statement at the [edit interfaces ls-*fpc*/*pic*/*port* mlfr-uni-nni-bundle-options] hierarchy level:

```
[edit interfaces ls-fpc/pic/port mlfr-uni-nni-bundle-options]
acknowledge-retries number;
```

Acknowledgement retries can be a value from 1 through 5. The default is 2.

You can configure the rate at which hello messages are sent by including the hello-timer statement at the [edit interfaces ls-*fpc*/*pic*/*port* mlfr-uni-nni-bundle-options] hierarchy level:

```
[edit interfaces ls-fpc/pic/port mlfr-uni-nni-bundle-options]
hello-timer milliseconds;
```

A hello message is transmitted after the specified period (in milliseconds) has elapsed. The hello timer can be from 1 through 180 milliseconds; the default is 10 milliseconds. When the hello timer expires, a link endpoint generates an add link message.

Configure Link Services Differential Delay

For link services interfaces only, the differential delay between links in a bundle is measured and warning is given when a link has a substantially greater differential delay than other links in the same bundle. The implementing endpoint can determine if the differential delay is in an acceptable range and decide to remove the link from the bundle, or to stop transmission on the link.

You can configure the yellow differential delay for links in a bundle by including the yellow-differential-delay statement at the [edit interfaces ls-*fpc*/*pic*/*port* mlfr-uni-nni-bundle-options] hierarchy level:

```
[edit interfaces ls-fpc/pic/port mlfr-uni-nni-bundle-options]
yellow-differential-delay milliseconds;
```

The yellow differential delay can be from 1 through 2000 milliseconds. The default is 6 milliseconds.

You can configure the red differential delay for links in a bundle to give warning by including the red-differential-delay statements at the [edit interfaces ls-*fpc*/*pic*/*port* mlfr-uni-nni-bundle-options] hierarchy level:

```
[edit interfaces ls-fpc/pic/port mlfr-uni-nni-bundle-options]
red-differential-delay milliseconds;
```

The red differential delay can be from 1 through 2000 milliseconds. The default is 10 milliseconds.

You can configure the action to be taken when differential delay exceeds the red limit by including the action-red-differential-delay red statements at the [edit interfaces ls-*fpc*/*pic*/*port* mlfr-uni-nni-bundle-options] hierarchy level:

```
[edit interfaces ls-fpc/pic/port mlfr-uni-nni-bundle-options]
action-red-differential-delay (disable-tx | remove-link);
```

The disable-tx option disables transmission on the link. The remove-link option removes the link from the bundle. The default action is disable-tx.

Configure Link Services Keepalive Settings on Frame Relay LMI

You can tune the keepalive settings on the physical link-services interface. By default, the JUNOS software uses Q.933 Annex A LMIs for FRF.16. To use ITU Annex A LMIs, include the lmi-type itu statement at the [edit interfaces ls-*fpc*/*pic*/*port* mlfr-uni-nni-bundle-options] hierarchy level:

```
[edit interfaces ls-fpc/pic/port mlfr-uni-nni-bundle-options]
lmi-type itu;
```

To configure Frame Relay keepalive parameters on a link services interface, include the n391, n392, n393, t391 and t392 statements at the [edit interfaces ls-*fpc*/*pic*/*port* mlfr-uni-nni-bundle-options] hierarchy level:

```
[edit interfaces ls-fpc/pic/port mlfr-uni-nni-bundle-options]
n391 number;
n392 number;
n393 number;
t391 number;
t392 number;
```

You can set the following properties:

n391—Full status polling interval. The data terminal equipment (DTE) sends a status inquiry to the data communication equipment (DCE) at the interval specified by t391. n391 specifies the frequency with which these inquiries expect a full status report; for example, an n391 value of 10 would specify a full status report in response to every tenth inquiry. The intermediate inquiries ask for a keepalive exchange only. The range is 1 through 255, with a default value of 6.

n392—Error threshold. The number of errors required to bring down the link, within the event count specified by n393. The range is 1 through 10, with a default value of 3.

n393—Monitored event count. The range is 1 through 10, with a default value of 4.

t391—Keepalive timer. Interval at which the DTE sends out a keepalive response request to the DCE and updates status, depending on the error threshold value. The range is 5 through 30 seconds, with a default value of 10 seconds.

t392—Keepalive timer. Period during which the DCE checks for keepalive responses from the DTE and updates status, depending on the DCE error threshold value. The range is 5 through 30 seconds, with a default value of 15 seconds.

> **Note**
>
> For the LMI to work properly, you must configure one side of a link services bundle to be a DCE.

Multilink and Link Services Interface Structure

Each Multilink Services or Link Services PIC can support a number of *bundles* . A bundle can contain up to eight individual *links*. For Multilink Services PICs, the links can be T1 or E1 physical interfaces, and each link is associated with a logical unit number that you configure. For Link Services PICs, the links can be E1, T1, Channelized DS-3 to DS-1, Channelized DS-3 to DS-0, Channelized E1, Channelized STM-1interfaces, or Channelized QPP interfaces. For link services bundles, each link is associated with a channel number that you configure.

You must configure a link before it can join a bundle. Each bundle should consist solely of one type of link; we recommend that you not mix physical interfaces within a bundle.

This section is organized as follows:

Multilink Services PIC Capacities

Three versions of Multilink Services PICs are available, as shown in Table 24. The PIC hardware is identical, except for different faceplates that enable you to identify which version you are installing. The software limits the unit numbers and maximum number of physical interfaces you assign to the PIC. You can install a maximum of four Multilink Services PICs per router.

Table 24: Multilink Services PIC Capacities

PIC Capacity	Unit Numbers	Maximum Number of T1 Interfaces	Maximum Number of E1 Interfaces
4-bundle PIC	0 through 3	32 links	32 links
32-bundle PIC	0 through 31	256 links	219 links
128-bundle PIC	0 through 127	292 links	219 links

A single PIC can support an aggregate bandwidth of 450 Mbps.

You can configure a larger number of links, but the Multilink Services PIC can reliably process only 450 Mbps of traffic. A higher rate of traffic might degrade performance.

Link Services PIC Capabilities

The default number of bundles per Link Services PIC is 16, ranging from ls-*fpc*/*pic*/*port*:0 to ls-*fpc*/*pic*/*port*:15.

To configure the number of bundles on a Link Services PIC, include the mlfr-uni-nni-bundles statement at the [edit chassis fpc *slot-number* pic *pic-number*] hierarchy level:

```
[edit chassis fpc slot-number pic pic-number]
mlfr-uni-nni-bundles number;
```

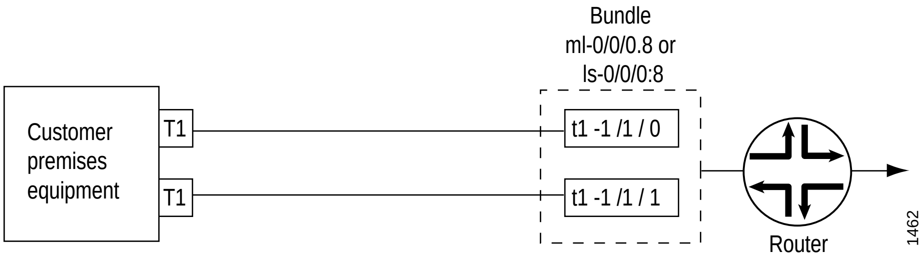
The maximum number of MLFR UNI NNI bundles each Link Services PIC can accommodate is 128. For more information, see the *JUNOS Internet Software Guide: Getting Started*.

A link can associate with one link services bundle only. For an example configuration, see “Examples: Configure Link Services Interfaces” on page 288.

Configure Multilink and Link Services Bundles

To complete a multilink or link services interface configuration, you need to configure both the physical interface and the multilink or link services bundle. For multilink interfaces, you configure the link bundle on the logical unit. For link services interfaces, you configure the link bundle as a channel (see Figure 23). The physical interface is usually connected to networks capable of supporting MLPPP or MLFR (FRF.15 or FRF.16).

Figure 23: Multilink Interface Configuration



Using the topology in Figure 23 as an example, configure a multilink or link services bundle over a T1 connection (for which you have already configured the T1 physical interface) with the following additional configuration statements:

1. To configure a physical T1 link for MLPPP, include the following statements at the [edit interfaces t1-*fpc/pic/port*] hierarchy level:

   ```
   [edit interfaces t1-fpc/pic/port]
   unit 0 {
       family mlppp {
           bundle (ml-fpc/pic/port | ls-fpc/pic/port);
       }
   }
   ```

   You do not need to configure an IP address on this link.

   > **Note**
   >
   > If the MLPPP bundle is interoperating with Cisco IOS, the IOS configuration must include the ppp multilink fragment-delay *milliseconds* statement within the interface multilink# configuration group:
   >
   > ```
   > interface Multilink#
   > ppp multilink fragment-delay milliseconds
   > ```
   >
   > The delay should be 500 milliseconds.

   To configure a physical T1 link for MLFR (FRF.16), include the following statements at the [edit interfaces t1-*fpc/pic/port*] hierarchy level:

   ```
   [edit interfaces t1-fpc/pic/port]
   encapsulation multilink-frame-relay-uni-nni;
   unit 0 {
       dlci dlci-identifier;
       family mlfr-uni-nni {
           bundle ls-fpc/pic/port/channel;
       }
   }
   ```

   You do not need to configure an IP address on this link.

2. To configure the logical address for the MLPPP bundle, include the address and destination statements at the [edit interfaces ml-*fpc/pic/port* unit *logical-unit-number* family inet] hierarchy level:

   ```
   [edit interfaces ml-fpc/pic/port unit logical-unit-number]
       family inet {
           address address {
               destination address;
           }
       }
   }
   ```

   When you add statements such as MRRU to the configuration and commit, the T1 interface becomes part of the multilink bundle.

To configure the logical address for the MLFR (FRF.15) bundle, include the address and destination statements at the [edit interfaces ml-*fpc/pic/port* unit *logical-unit-number* family inet] hierarchy level:

```
[edit interfaces ml-fpc/pic/port unit logical-unit-number]
family inet {
    address address {
        destination address;
    }
}
```

To configure the logical address for the MLFR (FRF.16) bundle, include the address and destination statements at the [edit interfaces ls-*fpc/pic/port* unit *logical-unit-number* family inet] hierarchy level:

```
[edit interfaces ls-fpc/pic/port unit logical-unit-number]
encapsulation multilink-frame-relay-uni-nni;
family inet {
    address address {
        destination address;
    }
}
```

> **Note**
>
> For MLPPP and MLFR (FRF.15 and FRF.16) links, you must specify the subnet address as /32 or /30. Any other subnet designation is treated as a mismatch.

Examples: Configure Multilink Interfaces

These examples show only the multilink part of the configuration. To see the the T1 configuration options, see “Configure T1 Interfaces” on page 319.

Configure an MLPPP interface:

```
[edit interfaces]
ml-1/0/0 {
    unit 1 {
        fragment-threshold 128;
        family inet {
            address 192.128.5.1/32 {
                destination 192.128.200.200;
            }
        }
    }
    unit 10 {
        family inet {
            address 128.1.1.3/32 {
                destination 128.1.1.2;
            }
        }
    }
}
```

```
t1-5/1/0 {
    unit 0 {
        family mlppp {
            bundle ml-1/0/0.1;
        }
    }
}
t1-5/1/1 {
    unit 0 {
        family mlppp {
            bundle ml-1/0/0.1;
        }
    }
}
t1-5/1/2 {
    unit 0 {
        family mlppp {
            bundle ml-1/0/0.1;
        }
    }
}
```

Configure an MLFR (FRF.15) interface:

```
[edit interfaces]
ml-1/0/0 {
    unit 1 {
        encapsulation multilink-frame-relay-end-to-end;
        family inet {
            address 192.128.5.2/32 {
                destination 192.128.5.3;
            }
        }
    }
    unit 10 {
        encapsulation multilink-frame-relay-end-to-end;
        family inet {
            address 128.1.1.3/32 {
                destination 128.1.1.2;
            }
        }
    }
}
t1-5/1/0 {
        unit 0 {
            dlci 10;
            encapsulation multilink-frame-relay-end-to-end;
            family mlfr-end-to-end {
                bundle ml-1/0/0.1;
            }
    }
}
t1-5/1/1 {
    unit 0 {
        dlci 5;
        encapsulation multilink-frame-relay-end-to-end;
        family mlfr-end-to-end {
            bundle ml-1/0/0.10;
        }
    }
}
```

```
t1-5/1/2 {
    unit 0 {
        dlci 6;
        encapsulation multilink-frame-relay-end-to-end;
        family mlfr-end-to-end {
            bundle ml-1/0/0.10;
        }
    }
}
```

Examples: Configure Link Services Interfaces

This example shows only the link services part of the configuration. To see the the T1 configuration options, see “Configure T1 Interfaces” on page 319.

This example shows one bundle with two links, as listed in Table 25.

Table 25: Link Services Bundle

Router A	Router B
t1-0/1/0 (ls-1/1/0:3)	t1-0/3/0 (ls-0/0/0:10)
t1-0/1/1 (ls-1/1/0:3)	t1-0/3/1 (ls-0/0/0:10)

For LMI to work properly, you must configure one router to be a DCE.

On Router A:

```
[edit interfaces]
ls-1/1/0:3 {
    dce;
    encapsulation multilink-frame-relay-uni-nni;
    unit 0 {
        dlci 16;
        family inet {
            address 3.3.3.1/32 {
                destination 3.3.3.2;
            }
        }
    }
}

t1-0/1/0 {
    encapsulation multilink-frame-relay-uni-nni;
    unit 0 {
        family mlfr-uni-nni {
            bundle ls-1/1/0:3;
        }
    }
}
```

```
t1-0/1/1 {
    encapsulation multilink-frame-relay-uni-nni;
    unit 0 {
        family mlfr-uni-nni {
            bundle ls-1/1/0:3;
        }
    }
}
```

On Router B:

```
[edit interfaces]
ls-0/0/0:10 {
    encapsulation multilink-frame-relay-uni-nni;
    unit 0 {
        dlci 16;
        family inet {
            address 3.3.3.2/32 {
                destination 3.3.3.1;
            }
        }
    }
}

t1-0/3/0 {
    encapsulation multilink-frame-relay-uni-nni;
    unit 0 {
        family mlfr-uni-nni {
            bundle ls-0/0/0:10;
        }
    }
}

t1-0/3/1 {
    encapsulation multilink-frame-relay-uni-nni;
    unit 0 {
        family mlfr-uni-nni {
            bundle ls-0/0/0:10;
        }
    }
}
```

This configuration initiates the MLFR UNI NNI protocol between Router A and Router B and logically connects link services bundles ls-1/1/0.3 and ls-0/0/0.10.

The following three examples show configuration of Link Services interfaces with MLPPP, MLFR FRF.15, and MLFR FRF.16.

Configure a Link Services interface with MLPPP:

```
[edit interfaces]
t1-0/0/0 {
    encapsulation ppp;
    unit 0 {
        family mlppp {
            bundle ls-0/3/0.0;
        }
    }
}
t1-0/0/1 {
    encapsulation ppp;
    unit 0 {
        family mlppp {
            bundle ls-0/3/0.0;
        }
    }
}
ls-0/3/0 {
    unit 0 {
        encapsulation multilink-ppp;
        family inet {
            address 10.16.1.2/32 {
                destination 10.16.1.1;
            }
        }
        family iso;
        family inet6 {
            address 8016::1:2/126;
        }
    }
}
```

Configure a Link Services interface with MFR FRF.15:

```
[edit interfaces]
t1-0/0/0 {
    encapsulation frame-relay;
    unit 0 {
        dlci 16;
        family mlfr-end-to-end {
            bundle ls-0/3/0.0;
        }
    }
}
t1-0/0/1 {
    encapsulation frame-relay;
    unit 0 {
        dlci 16;
        family mlfr-end-to-end {
            bundle ls-0/3/0.0;
        }
    }
}
```

```
ls-0/3/0 {
    unit 0 {
        encapsulation multilink-frame-relay-end-to-end;
        family inet {
            address 10.16.1.2/32 {
                destination 10.16.1.1;
            }
        }
        family iso;
        family inet6 {
            address 8016::1:2/126;
        }
    }
}
```

Configure a Link Services interface with MLFR FRF.16:

```
[edit chassis]
fpc 1 {
    pic 2 {
        mlfr-uni-nni-bundles 5;
    }
}

[edit interfaces]
t1-0/0/0 {
    encapsulation multilink-frame-relay-uni-nni;
    unit 0 {
        family mlfr-uni-nni {
            bundle ls-1/2/0:0;
        }
    }
}
t1-0/0/1 {
    encapsulation multilink-frame-relay-uni-nni;
    unit 0 {
        family mlfr-uni-nni {
            bundle ls-1/2/0:0;
        }
    }
}
ls-1/2/0:0 {
    dce;
    encapsulation multilink-frame-relay-uni-nni;
    unit 0 {
        dlci 26;
        family inet {
            address 10.26.1.1/32 {
                destination 10.26.1.2;
            }
        }
    }
}
```